**НЕТОЛОГИЯ**

**Дипломный проект**

**«Прогнозирование продаж фармдистрибьютора»**

Работу выполнила:Федосеева Мария

Профессия: Data Science (DS-16)

Москва, 2021

**Постановка задачи**

В данной работе рассмотрены продажи компании, которая занимается дистрибьюцией лекарственных препаратов. Рынок очень насыщенный и конкурентный, поэтому идет борьба за каждый процент его доли. Чтобы эффективно конкурировать и как минимум не отставать, нужно прежде всего обладать базовыми знаниями о будущем движении рынка. Современные методы предективной аналитики способный предсказать тренды и исходя из этого можно спрогнозировать необходимые ресурсы. Так, если намечается рост объемов продаж, есть возможность подготовиться, например, нанять дополнительный персонал, закупить ресурсы, арендовать дополнительные складские площади и т.п. Если, наоборот, следует падение, то можно отказаться от дополнительного найма сотрудников, пересмотреть затраты, критически отнестись к открытию новых инвестиционных проектов.

Знание будущих трендов позволит грамотно выстроить стратегию развития компании, планировать необходимые ресурсы, повышать эффективность.

В рамках проекта будут рассмотрены два метода прогнозирования продаж на основе временных рядов, проведен их сравнительный анализ и сформулирован вывод.

Для работы возьмем статику продаж дистрибьютора за период с января 2019 года по июнь 2021 в разрезе дня и попробуем спрогнозировать будущие продажи в краткосрочной перспективе месяца.

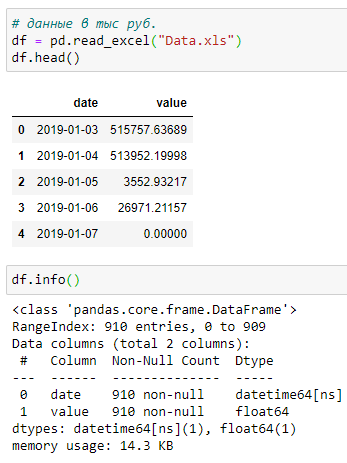
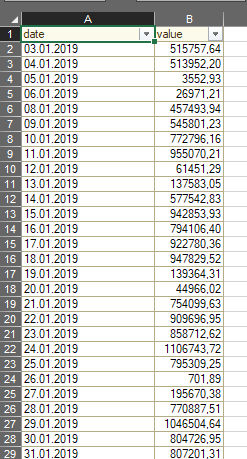
**Анализ**

На текущий момент существует множество моделей и подходов к прогнозированию временных рядов. Универсального и так называемой "серебряной пули" не существует, поэтому под каждую задачу и набор данных необходим свой подход.

В свободном доступе интернет можно ознакомится с аналогичными решениями.

* Например, хорошо исследована аналогичная тема в OZON [на странице](https://habr.com/ru/company/ozontech/blog/431950/) достаточно подробное решение и в итоге получен результат с MAE = 1.
* Еще [как пример](https://www.machinelearningmastery.ru/predicting-sales-611cb5a252de/) описано построение модели LSTM, которая тоже показала неплохие результаты предсказания на 6 месяцев.

Итак, посмотрим на датасет:

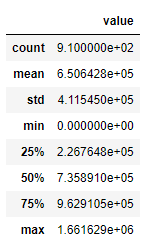
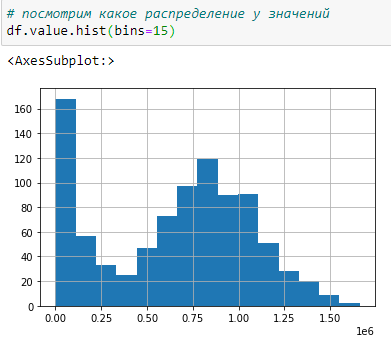
 

Датасет содержит две колонки:

Date – дата продажи

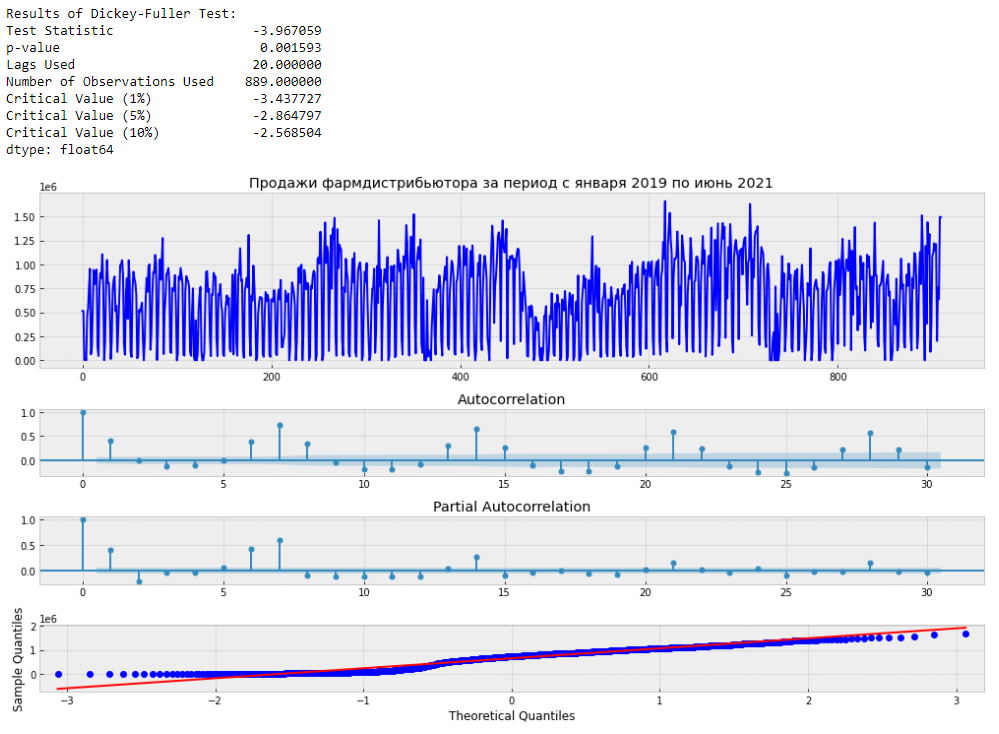
Value – сумма продаж с НДС на дату, в тыс. рублей.

Данных очень мало, за 2,5 года всего 910 значений. При этом нет пропусков в датах. Значение value с плавающей точкой. В категории миллионов для удобства округлим значение до целочисленного и посмотрим основную статистику.

По статистике видно, что дистрибьютор реализовывает товара в среднем в день на 650 642,8 тыс. руб., отрицательных значений в данных нет, есть дни без продаж, о чем говорит минимальное значение, равное 0, максимальная сумма продаж в день с 2019г равна 1 661 629 тыс. руб. Никаких аномалий и выбросов по статистике не замечено. Распределение значений продаж не является нормальным.

Вначале необходимо посмотреть на коррелограммы и тест Дики-Фулера ряда, чтобы понять, что из себя представляют данные.



Значение «p-value» равно 0,001593, что меньше 0,05, а значит тест отвергает гипотезу о нестационарности ряда. Ряд стационарен.

На графике автокорреляции прослеживается сезонность, примерно 7 периодов. Явный тренд отсутствует, цикличности тоже нет.

В целом на графике остатков нормальное распределение, но есть небольшие дефекты.

В исследовании будем сравнивать результаты моделей SARIMA, так как присутствует сезонность, и рекуррентной нейронной сети LSTM. Можно было бы рассмотреть и линейную регрессию, как универсальное средство, но хотелось бы сосредоточиться именно на моделях для прогноза временных рядов.

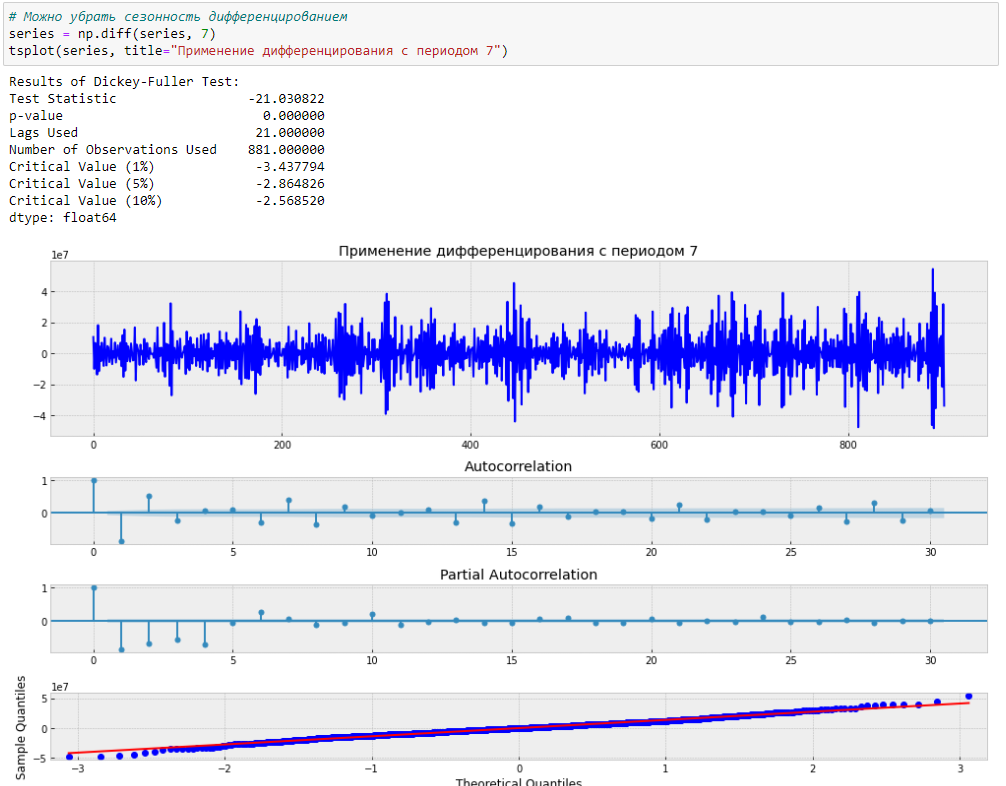
**Методика решения**

В исследуемом ряду нет дисперсии (разброс значений случайной величины), но если бы была, мы нормализовали ее при помощи преобразования Бокса-Кокса, например:

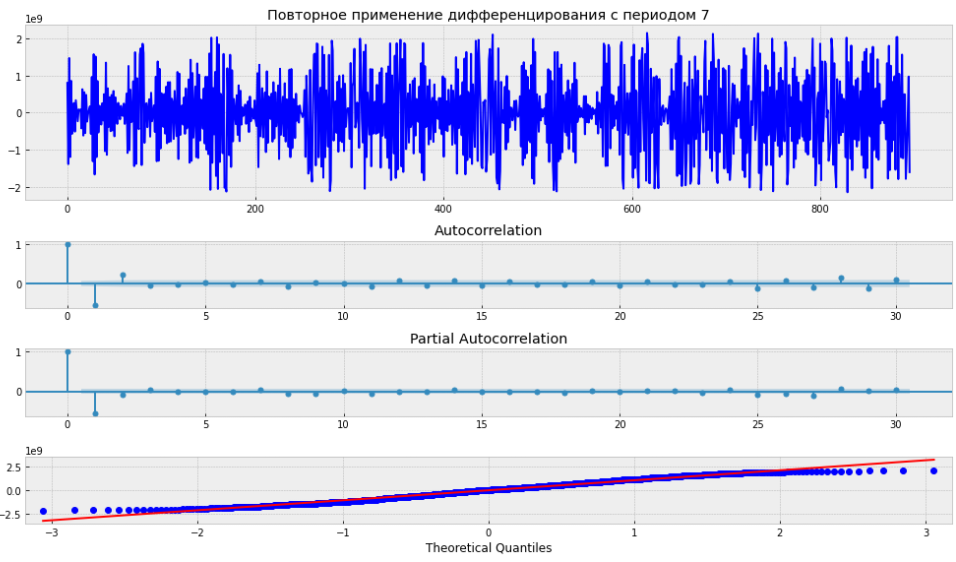
*from scipy.stats import boxcox*

*series = boxcox(series, 0)*

Если бы сезонность играла сильную роль в исследуемых моделях мы бы убрали ее дифференцированием следующим образом.



При этом мы достигаем в тесте Дики-Фуллера значение p-value равное 0, и как видим, изменилось распределение остатков в лучшую сторону. Но на графике автокорреляции все еще есть сезонность. Применим дифференцирование повторно с тем же самым периодом.



После повторного применения виден отличный результат на графиках, ряд очевидно стационарен, убрана сезонность, остатки нормально распределены. Если бы в работе рассматривалась в качестве примера линейная регрессия, то применение дифференцирования было бы необходимым действием, но для SARIMA и LSTM в данной работе убирать сезонность не требуется, поэтому будем рассматривать исходные данные.

Для оценки качества моделей будем использовать метрику RSME (квадратный корень из среднеквадратической ошибки)

Для вычисления среднеквадратической ошибки (MSE) все отдельные [остатки регрессии](http://statistica.ru/glossary/general/ostatki-regressii/) возводятся в квадрат, суммируются, сумма делится на общее число ошибок:

MSE

Квадратный корень из данной величины обозначается как RMSE (Root Mean Square Error):

RMSE

**SARIMA**

**Авторегрессионное интегрированное скользящее среднее (autoregressive integrated moving average, ARIMA)** является обобщением модели [авторегрессионного скользящего среднего](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%81%D0%BA%D0%BE%D0%BB%D1%8C%D0%B7%D1%8F%D1%89%D0%B5%D0%B5_%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B5%D0%B5" \o "Авторегрессионное скользящее среднее). Эти модели используются при работе с [временными рядами](http://www.machinelearning.ru/wiki/index.php?title=%D0%92%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9_%D1%80%D1%8F%D0%B4) для более глубокого понимания данных или предсказания будущих точек ряда. Обычно модель упоминается, как ARIMA(p,d,q), где p,d и q — целые неотрицательные числа, характеризующие порядок для частей модели (соответственно авторегрессионной, интегрированной и скользящего среднего).

Пусть задан временной ряд X_t, где t — целый индекс и X_t — вещественные числа. Тогда модель ARMA(p,q) задаётся следующем образом:

\left(1-\sum_{i=1}^p \phi_i L^i\right) X_t = \left(1+\sum_{i=1}^q \theta_i L^i\right) \epsilon_t,

где L — оператор задержки, \phi_i — параметры авторегрессионной части модели, \theta_i — параметры скользящего среднего, а \epsilon_t — значения ошибки. Обычно предполагают, что ошибки \epsilon_t являются независимыми одинаково распределёнными случайными величинами из нормального распределения с нулевым средним.

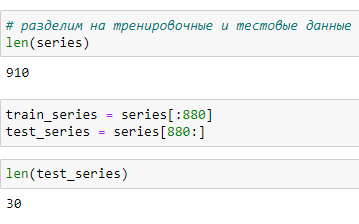
ARIMA(p,d,q) получается интегрированием ARMA(p,q).

\left(1-\sum_{i=1}^p \phi_i L^i\right) (1-L)^d X_t = \left(1+\sum_{i=1}^q \theta_i L^i\right) \epsilon_t,

где d — положительное целое, задающее уровень дифференцирования (если d=0, эта модель эквивалентна авторегрессионному скользящему среднему). И наоборот, применяя почленное дифференцирование d раз к модели ARMA(p,q), получим модель ARIMA(p,d,q). Заметим, что дифференцировать надо только авторегрессионную часть.

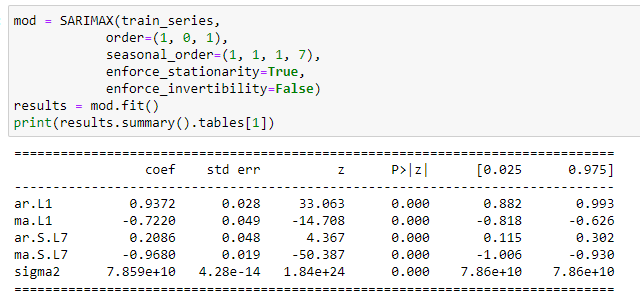
**Сезонное авторегрессионное интегрированное скользящее среднее, SARIMA или Seasonal ARIMA**, является расширением ARIMA, которое явно поддерживает одномерные данные временных рядов с сезонным компонентом. Оно добавляет три новых гиперпараметра для указания авторегрессии (AR), разности (I) и скользящего среднего (MA) для сезонной составляющей ряда, а также дополнительный параметр для периода сезонности. SARIMA (p, d, q), (P, D, Q)S , где (p, d, q) – несезонная часть и, соответственно, (P, D, Q) является сезонной частью модели.

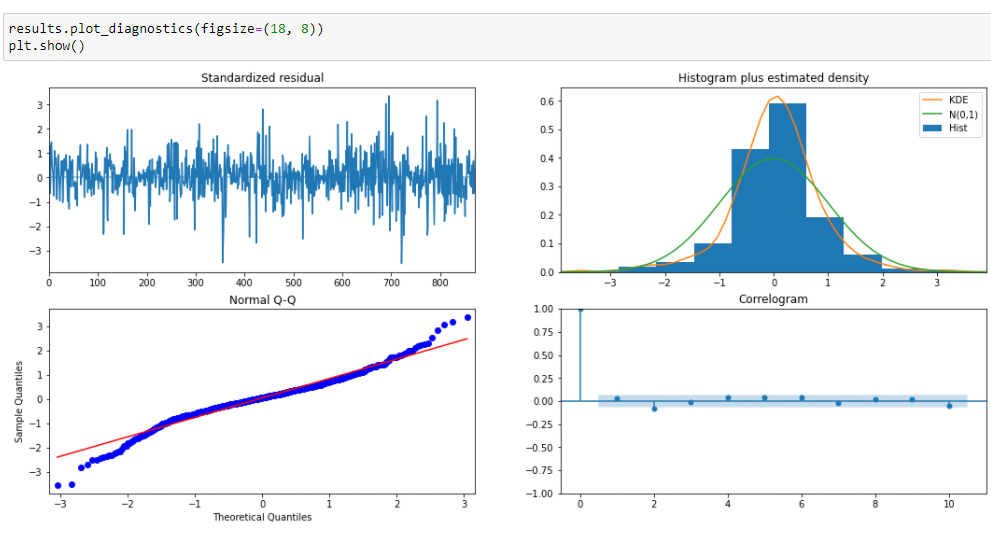
Разделим данные на тестовые и тренировочные и подберём оптимальные значения гиперпараметров по критерию Акаике.

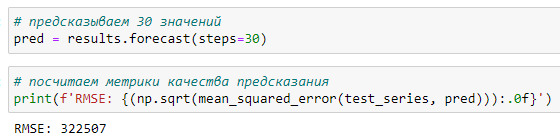


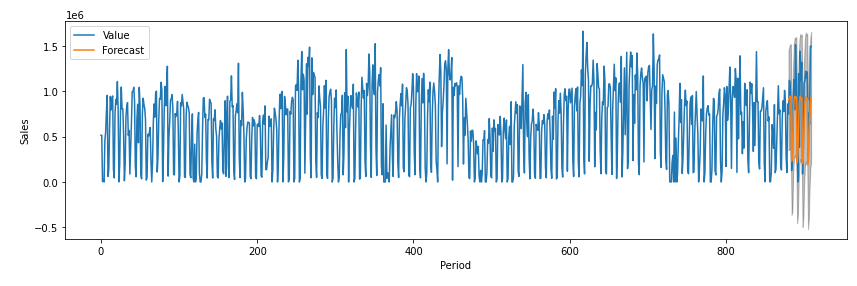


Параметры, которые описывают ряд (1, 0, 1)х(1, 1, 1, 7)7. Используем их для создания и обучения модели на тренировочных данных. Затем предскажем 30 значений периода и посмотрим на качество модели.





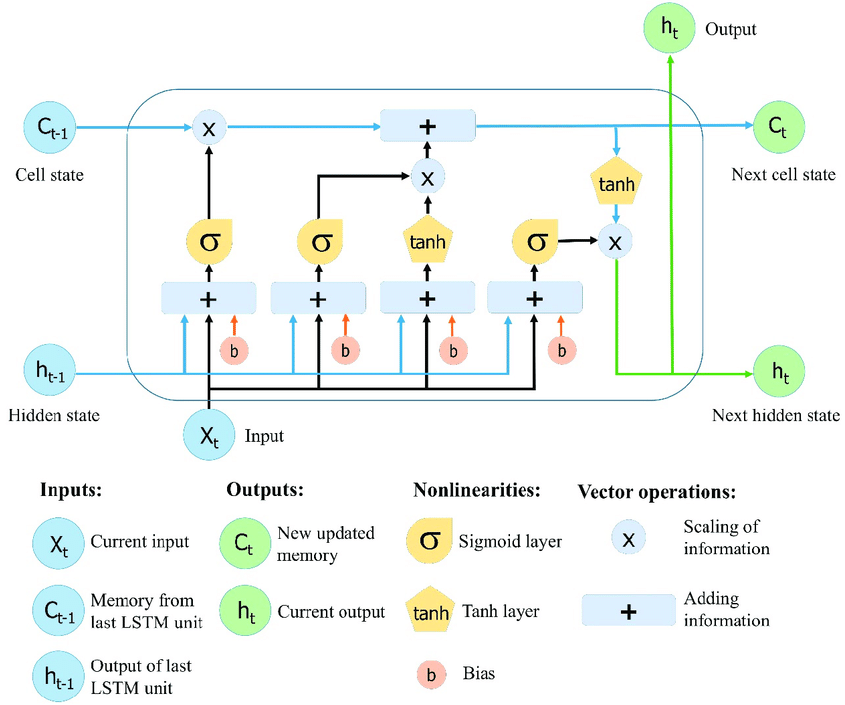




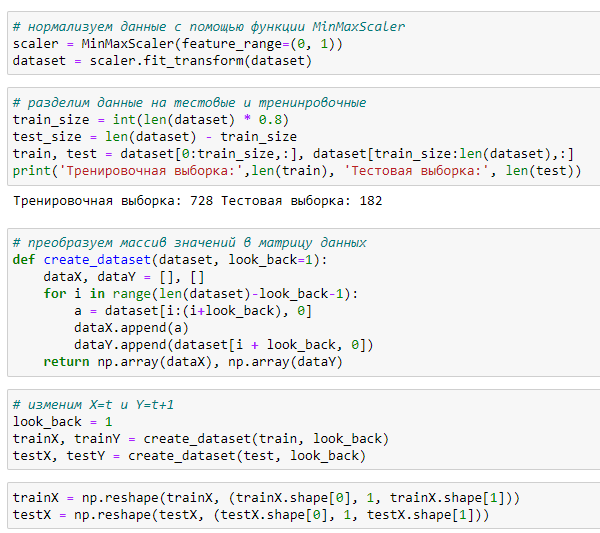
RMSE SARIMA: 322507

**LSTM**

**Долгая краткосрочная память (**[**англ.**](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)**Long short-term memory; LSTM)** — разновидность архитектуры [рекуррентных нейронных сетей](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), предложенная в [1997 году](https://ru.wikipedia.org/wiki/1997_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BD%D0%B0%D1%83%D0%BA%D0%B5) [Зеппом Хохрайтером](https://ru.wikipedia.org/w/index.php?title=%D0%A5%D0%BE%D1%85%D1%80%D0%B0%D0%B9%D1%82%D0%B5%D1%80,_%D0%97%D0%B5%D0%BF%D0%BF&action=edit&redlink=1" \o "Хохрайтер, Зепп (страница отсутствует)) и [Юргеном Шмидхубером](https://ru.wikipedia.org/wiki/%D0%A8%D0%BC%D0%B8%D0%B4%D1%85%D1%83%D0%B1%D0%B5%D1%80,_%D0%AE%D1%80%D0%B3%D0%B5%D0%BD)[[2]](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%B3%D0%B0%D1%8F_%D0%BA%D1%80%D0%B0%D1%82%D0%BA%D0%BE%D1%81%D1%80%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C#cite_note-lstm1997-2). Как и большинство рекуррентных нейронных сетей, LSTM-сеть является [универсальной](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B0_%D0%BF%D0%BE_%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3%D1%83) в том смысле, что при достаточном числе элементов сети она может выполнить любое вычисление, на которое способен обычный компьютер, для чего необходима соответствующая [матрица](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) весов, которая может рассматриваться как программа. В отличие от традиционных рекуррентных нейронных сетей, LSTM-сеть хорошо приспособлена к обучению на задачах [классификации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8), обработки и [прогнозирования](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [временных рядов](https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9_%D1%80%D1%8F%D0%B4) в случаях, когда важные события разделены временными лагами с неопределённой продолжительностью и границами.

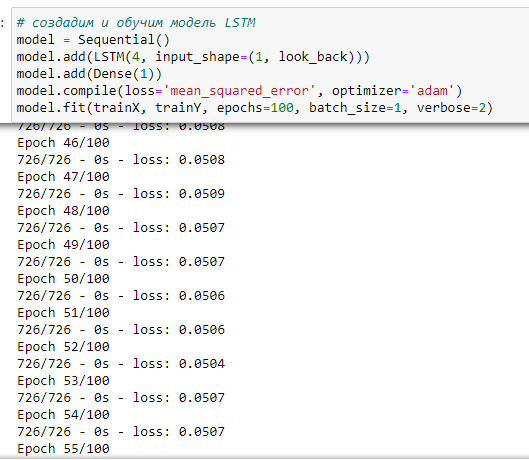
****

Аналогично предыдущей модели разделим данные на тренировочные и тестовые, предварительно нормализуем их.

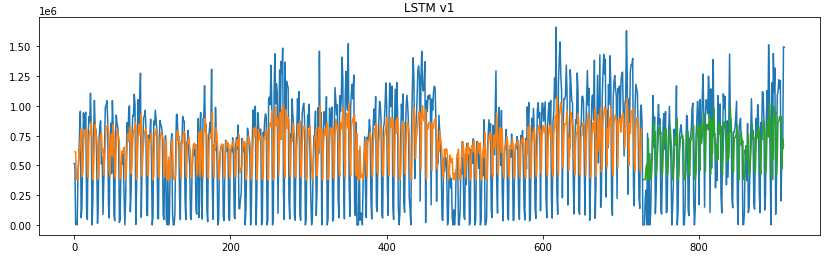


Создадим и обучим модель LSTM. Модель имеет 4 слоя, в качестве функции потерь будем использовать MSE и оптимизатор Adam, как наиболее оптимальные параметры модели. [Источник информации](https://deepdatascience.wordpress.com/2016/11/18/which-lstm-optimizer-to-use/)

***LSTM. Вариант 1.***

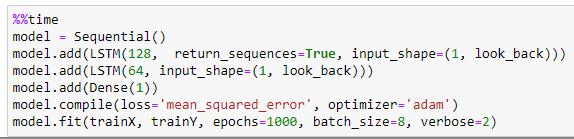


Итоговый график предсказания модели v1:

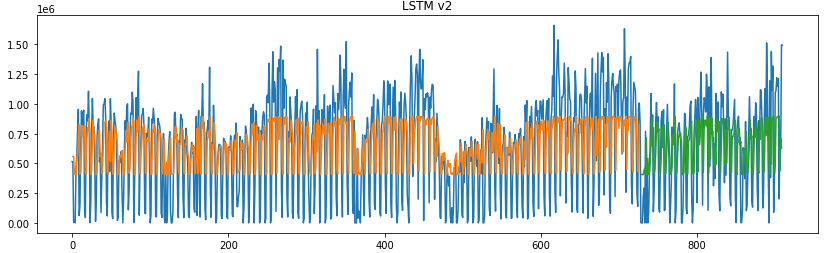


*RMSE LSTM v1: 376719*

***LSTM. Вариант 2. Добавим эпох, увеличим батчи, добавим слои.***

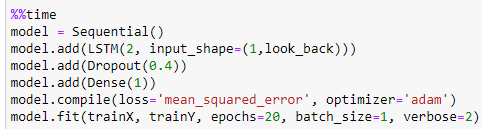


Итоговый график предсказания модели v2:

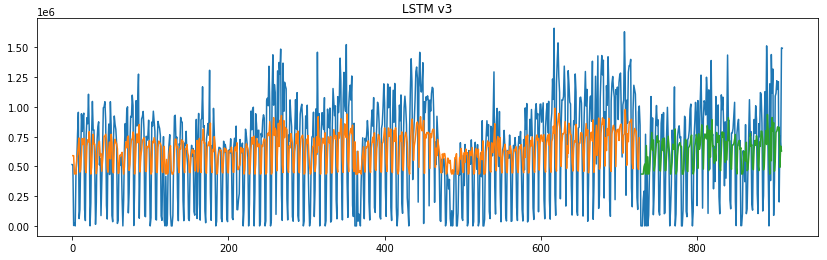


*RMSE LSTM v2: 381482*

***LSTM. Вариант 3. Добавим слой Dropout.***



Итоговый график предсказания модели v3:



*RMSE LSTM v3: 375767*

**Выводы и заключение**

В данной дипломной работе было проведено обучение 2х моделей предсказания временного ряда на основе датасета ежедневных продаж фармдистрибьютора с 2019г по 2021г.

Исходя из небольшого размера первоначальных данных результат получился, на мой взгляд, средним. Наиболее успешной по критерию RSME стала модель SARIMA. Модели нейронной сети LSTM явно недостаточно данных для обучения и даже при разных трех вариантах модификации сезонная авторегрессионная скользящая показывает лучший результат.

|  |  |
| --- | --- |
| **Модель** | **RMSE** |
| SARIMA | 322507 |
| LSTM v1 | 376719 |
| LSTM v2 | 381482 |
| LSTM v3 | 375767 |

В дальнейшем можно улучшить результаты каждой из модели, если взять бОльший период исходных данных. Взять, например, не дневной диапазон, а месячный. Разбить данные продаж по подразделениям (оптовые, аптечные продажи, бюджетные и т.п.) и рассмотреть их отдельно.

**Список источников**

1. [Wiki LSTM](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%B3%D0%B0%D1%8F_%D0%BA%D1%80%D0%B0%D1%82%D0%BA%D0%BE%D1%81%D1%80%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C)
2. [Прогнозирование временных рядов с помощью рекуррентных нейронных сетей LSTM в Python с использованием Keras](https://www.machinelearningmastery.ru/time-series-prediction-lstm-recurrent-neural-networks-python-keras/)
3. [LSTM — нейронная сеть с долгой краткосрочной памятью](https://neurohive.io/ru/osnovy-data-science/lstm-nejronnaja-set/)
4. [Wiki ARIMA](http://www.machinelearning.ru/wiki/index.php?title=ARIMA)
5. [SARIMA](https://www.machinelearningmastery.ru/sarima-for-time-series-forecasting-in-python/)
6. [LSTM Optimizer Choice ?](https://deepdatascience.wordpress.com/2016/11/18/which-lstm-optimizer-to-use/)
7. [Линейная регрессия в машинном обучении](https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/)